METHOD AND APPARATUS FOR DOWNSCALING A DIGITAL MATRIX IMAGE

The invention relates to a method and apparatus for downscaling a digital matrix image using a selected ratio, in which the matrix image includes a large number of lines, each line including a large number of pixels, so that the intensity values of the pixels form the matrix, and in which the output matrix pixels formed by scaling correspond to the sub-groups of the original matrix, from the intensity values of the pixels of which an average is calculated in a selected manner for each pixel of the output matrix.

Camera sensors are used when using digital cameras to take individual images or video images. The image of the sensor can use various image formats, for example, RGB8;8:8, RGB5:6:5, YUV4:2:0 and a raw-Bayer image. When the image is shown in the viewfinder (VF), which usually has a lower resolution than the image sensor, the image must be formed in the sensor and scaled to be suitable for the resolution of the display. The image can also be zoomed (a smaller image from the sensor is cropped and then scaled) to the viewfinder. In zooming, there should be many stages, so that the result of the zooming will appear to be continuous. When video images are taken, the resolution of the video image is usually lower than the resolution of the sensor. Similar scaling will therefore also be required when videoing. Camera sensors can also be used equally well in portable devices as in cameras. Image scaling is needed, if the viewfinder image is running in real time on the display of a telephone or camera, or if a video image is being taken in real time.

Methods are known, in which the image is scaled (sub-sampled) using a low quality algorithm in the camera (poor image quality). The poorer quality is especially visible in the digital zoom mode of some DSCs (DSC = Digital Still Camera).

US publication 6,205,245 discloses one method, in which a colour image is scaled directly from the matrix of the sensor, in such a way that a pixel group, which is always processed at one time, is

defined, more or less corresponding to each pixel of the final image.

Generally, the scaling of an image matrix M_1 x N_1 is scaled to a smaller size M_2 x N_2 as follows. The scaling ratios M_2/M_1 and N_2/N_1 determine the procedure in the calculation operation. If scaling takes place in real time, i.e. As a continuous flow, memory need not be reserved for the input matrix, only three memory lines being sufficient. Consider the data coming in the X lines. The first memory line sums the amount according to the scaling ration in the X direction at the same time as the value of each pixel is summer in the Y line memory. If the scaling ratio results in the number of pixels not being an integer, the value of the pixel at the limit is weighted and summed with the two adjacent input pixels. In the same way, pixel values are calculated according to the scaling ration into the Y line memory and, in the case of boundary pixels, they are divided weighted into two parts. When the counter set for the Y scaling ration shows that the Y line memory is full, it is emptied forwards, after which summing starts from the beginning.

If the scaling ratios are small (near to zero), several memory lines will be needed, but their size is small. If, on the other hand, the scaling ratios are large (near to unity), only a few memory lines will be needed, but their size is large. Thus, depending on the scaling ratio, a fairly constant amount of memory is needed, the amount of which is about 3 x M_1 for a single colour component. In a practical application, the whole memory needed for scaling is less than 4 x M_1 for a single colour component.

The invention is intended to improve the quality of the viewfinder or display image, by using separate scaling ratios and smooth zooming. In addition, it is desired to reduce the level of interference. By means of the invention, it is wished to process an image with nearly optimum quality, while simultaneously keeping the demands for memory and power consumption at an economical level.

The characterizing features of the method and apparatus according to the invention are stated in the accompanying Claims. The invention is

particularly suitable for hardware-based (HW) implementations. The quality of scaling is nearly optimal and the level of interference is reduced significantly.

By means of the invention, the following advantages are achieved:

- Permits high-resolution images to be downscaled to both a display and a videocoder.
- Minimizes amount of memory required, despite high-quality downscaling.
  - ❖ High-frequency aliasing eliminated
    - lines and edges are represented correctly (no broken lines or jagged edges)
    - flickering of sharp lines and glimmering of high-contrast details eliminated.
- Noise level of output image attenuated.
  - ❖ Images can also be captured in dim/night conditions.
- Minimization of processing power requirement for high-quality images.

The quality of scaling is affected by the coarse and fine-scaling ratios. Quality is reduced if the pixels that are ready averaged in coarse scaling do not coincide with the fine-scaling limit, in which case they contain information from outside the output pixel.

In the following, the invention is examined with reference to the accompanying figures, which show some embodiments of the invention.

Figure 1 shows the solution in principle of the method according to the invention,
Figure 2a shows a block diagram of one device solution,
Figure 2b shows a block diagram on a second device solution,
Figure 2c shows the scaling solution of Figure 2b at a circuit level,
Figure 3 shows a horizontal depiction of 5/8-scaling,
Figure 4a shows a diagram of the division of the total ratio between the first and the second stages, when seeking to minimize calculation,

Figure 4b shows a diagram of the division of the total ratio between the first and second stages, when seeking to minimize the amount of memory required,

Figure 4c shows a diagram of the division of the total ratio between the first and second stages, when seeking to optimize mainly the quality of the image, but also the amount of calculation and memory required.

The method according to the invention includes two scaling stages, Figure 1. The first, coarse stage is simple and may comprise only the ratio 1/X. The next stage (fine) is more flexible, and may comprise the ratios Y/Z, in which Y < Z. X, Y, and Z are integers. The total scaling ration is the result of the scaling ratio in both stages. The smaller the first scaling ratio (Note: 1/3 < 1/2), the less memory will be required in the second stage. A smaller scaling ration in the first stage will also reduce the computational logistics and the total number of calculations. The first stage can be shown in an analog or digital form. The second stage defines the memory requirement. If the scaling ration is not directly 1/X, a better image quality will be achieved by using a smaller ratio in the second stage, but this will demand a larger memory.

The invention is particularly suitable for hardware-based applications, an example of which is the application according to Figure 2a. The camera module 10 is connected to a host system 22, which controls the display device 24 and the camera module. The camera module 10 includes particularly optics, i.e. a lens arrangement 11 (in practice, several lens), a sensor 12, an image-processing circuit 14, a scaling unit 16, and a controller 20. The image-processing circuit 14 reads, in a known manner, the sensor 12, thus creating a rapid data flow, which is led to the scaling unit 16, from which scaled data flow representing the selected image area is led to the host system 22. In the scaling unit 16, the data flow is first processed in a coarse scaler 17, from which the data flows relating to the intermediate image are led to a fine scaler 18, which makes the final scaling.

In one embodiment, the input and output units are separate units, due to the large data flow, each scaler having its own CPU and memory area on the same chip (not shown).

In Figure 2b, the same reference numbers as in Figure 2a are used for components that are functionally similar. In the solution of the figure, the actual camara module is slightly simpler, as the fine scaling 18 has been moved to the most system 22. The scaler 16' of the camera module includes only a coarse scaler 17. In this case the memory requirement is half a line in the coarse scaler (in the camera module) and three lines in the fine scaler (in the host module). For a sensor 1152x864, one line of memory represents Cx1152 words, in which C is the number of colour components (generally 3 - for RGB or YUV images). The length of the word depends on the accuracy of the calculation and is, for example, 2 or 4 bytes.

In one (fully digital embodiment) the construction of the scaler at the circuit level is according to Figure 2c. The coarse scaler 16' includes an input unit 161, a CPU 162, a memory 163, an output unit 164, and a controller 167, connected to an internal bus 165. The output unit 164 of this is connected to the input unit 181 of the fine scaler 18 (in the host system 22). The construction of the fine scaler 18 is similar, including the following components connection to a common bus 185: CPU 182, memory 183, output unit 184 and controller 187.

In this embodiment, scaling is performed using integers, which is much simpler to implement on a chip than floating-point calculation.

Figure 3 shows the significance of the variables MAXSTEP and PIXEL-STEP in the second scaler 3. The example shows the horizontal stage, but scaling is applied in both directions.

The values of an output pixel can be calculated as follows from the pixels of the intermediate image (Fig. 3):

$$A = (P2(a) * a + P1(b) * b) / 256$$

$$B = (P2(b) * b + P(c) * c + P1(d) * d) / 256$$

*C = (P2(d) * d + P1(e) * e) / 256*

*D = (P2(e) * e + P(f) * f + P1(g) * g) / 256*

*E = (P2(g) * g + P1(h) * h) / 256*

The PIXELSTEP value can be defined for the scaling ratio 5/8:

Set MAXSTEP = 256

*Tmp2 = (MAXSTEP * 5) / 8 = 160*

PIXELSTEP = floor(Tmp2) = 160

The weighting coefficients can be defined as follows:

P1(a) = 0

P1(x) = MAXSTEP - P2(x-1)

Conditional statement If (P1(x) > PIXELSTEP then

P1(x+1) = P1(x) - PIXELSTEP and P(x) = PIXELSTEP

P2(x) = PIXELSTEP - P1(x)

And so that weighting coefficients of the example given above can be calculated :

P1(a) = 0

P2(a) = PIXELSTEP - P1(a) = 160 - 0 = 160

P1(b) = MAXSTEP - P2(a) = 256 - 160 = 96 <= 160 96

P2(b) = PIXELSTEP - P1(b) = 160 - 96 = 64

P1(c) = MAXSTEP - P2(b) = 256 - 64 = 192 > 160

P(c) = PIXELSTEP = 160

P1(d) = P1(c) -PIXEL STEP = 192 - 160 = 32

P2(d) = PIXELSTEP - P1(d) = 160 - 32 = 128

P1(e) = MAXSTEP - P2(d) = 256 - 128 = 128 <= 160 128

P2(e) = PIXELSTEP - P1(e) = 160 - 128 = 32

P1(f) = MAXSTEP - P2(e) = 256 - 32 = 224 > 160

P(f) =PIXELSTEP = 160

P1(g) = P1(f) - PIXELSTEP = 224 - 160 = 64

P2(g) = PIXELSTEP - P1(g) = 160 - 64 = 96

P1(h) = MAXSTEP - P2(g) = 256 - 96 = 160 <= 160 160

Note! P2(h) = PIXELSTEP - P1(h) = 160 - 160 = 0

P1(i) = MAXSTEP - P2(h) = 256 - 0 > 160

P(i) = 160, P1(j) = 96

In the case of Figure 4a, the first stage scales the image as far as possible, using the ratio 1/X, in which X is an integer. The second stage carries out fine scaling, using the smallest possible amount of memory and minimum calculation. This means that the second scaling ratio is as large as possible (between [1/2, 1]) and thus three lines of memory are required.

The calculation is performed using the most advantageous integers. In the examples, the following concepts are used

- total scaling ratio Y/(X * Z) - marked SCRatio
- inverted total scaling ratio 1/SCRatio - marked IR
- function Floor() - take integer part (reject division remainder)
- function MAX()- select maximum value from list
- function Sqrt() - return square root
- function 2^() - return power of two
- logarithmic functions Log2() and Log10()
- auxiliary variables AVESKIP and PIXELSTEP, which are defined in the following:

AVESKIP:

IR = MAX(Hin / Hout, Vin / Vout), in which horizontal (H) and vertical (V) sizes are used.

AVESKIP = Floor(IR)

PIXELSTEP:

MAXSTEP = 256 (or 65536 if more precise pixel positioning is desired)

PIXELSTEP = Floor((MAXSTEP * AVESKIP)/IR)

Calculation example, scaling ratio (SCRatio) 0,182 i.e. ITR = 5,5:

AVESKIP = Floor(5,5) = 5

MAXSTEP = 256

PIXELSTEP = Floor (256 * 5 /5,5) = 232

In the case of Figure 4b, the first stage scales the image as much as possible, using the ratio 1/X, in which X is the power of two (2, 4, 8, 16, 64 etc.). The second stage carries out fine scaling, using as little memory as possible. This means that the scaling ration is between [1/2, 1] and thus three lines of memory are required.

In the following table, the stages of the scaling of Figure 4b are shown are numerical values. Scaling of this kind is used in, for example, zooming, in which the resolution of the initial image is 128 x 96. In the size of the part image is greater, it is scaled to this size. In the table, the original 1-megapixel image 1152 x 864 is scaled using the ratio 0,111 (1/8 x 128/144, index 64). In Figure 4b, the index value on the X-axis runs in the range 1 - 64 and the scaling ratio between 1,0 - 0,111.

X-size	Y-size	ratio X	X	Y	Z	index
128	96	1,000	1	128	128	1
132	99	0,970	1	128	132	2
137	103	0,934	1	128	137	3
141	106	0,908	1	128	141	4
146	110	0,877	1	128	146	5
151	114	0,848	1	128	151	6
157	118	0,815	1	128	157	7
162	122	0,790	1	128	162	8
168	126	0,762	1	128	168	9
174	131	0,736	1	128	174	10
180	135	0,711	1	128	180	11
187	140	0,684	1	128	187	12
193	145	0,663	1	128	193	13
200	150	0,640	1	128	200	14
208	156	0,615	1	128	208	15
215	161	0,595	1	128	215	16
222	167	0,577	1	128	222	17
230	173	0,557	1	128	230	18
237	177	0,540	1	128	237	19
246	184	0,520	1	128	246	20
256	192	0,500	2	128	128	21
264	198	0,485	2	128	132	22
274	206	0,467	2	128	137	23
282	212	0,454	2	128	141	24
292	220	0,438	2	128	146	25
302	228	0,424	2	128	151	26
314	236	0,408	2	128	157	27
324	244	0,395	2	128	162	28
336	252	0,381	2	128	168	29
348	262	0,368	2	128	174	30
360	270	0,356	2	128	180	31
374	280	0,342	2	128	187	32

X-size	Y-size	ratio X	X	Y	Z	index
386	290	0,332	2	128	193	33
400	300	0,320	2	128	200	34
416	312	0,308	2	128	208	35
430	322	0,298	2	128	215	36
444	334	0,288	2	128	222	37
460	346	0,278	2	128	230	38
474	354	0,270	2	128	237	39
492	368	0,260	2	128	246	40
512	384	0,250	4	128	128	41
528	396	0,242	4	128	132	42
548	412	0,234	4	128	137	43
564	424	0,227	4	128	141	44
584	440	0,219	4	128	146	45
604	456	0,212	4	128	151	46
628	472	0,204	4	128	157	47
648	488	0,198	4	128	162	48
672	504	0,190	4	128	168	49
696	524	0,184	4	128	174	50
720	540	0,178	4	128	180	51
748	560	0,171	4	128	187	52
772	580	0,166	4	128	193	53
800	600	0,160	4	128	200	54
832	624	0,154	4	128	208	55
860	644	0,149	4	128	215	56
888	668	0,144	4	128	222	57
920	692	0,139	4	128	230	58
948	708	0,135	4	128	237	59
984	736	0,130	4	128	246	60
1024	768	0,125	8	128	128	61
1056	792	0,121	8	128	132	62
1104	832	0,116	8	128	138	63
1152	864	0,111	8	128	144	64

In this case too, the calculation is performed using integers, so that in the example (Figure 3) the axiliary variables MAXSTEP, AVESKIP, and PIXELSTEP, which are defined in the following, are used:

AVESKIP:

Inverted total scaling ratio IR = MAX(Hin / Hout, Vin / Vout), in which horizontal (H) and vertical (V) sizes are used.

SKIP = Floor(Log2(IR))

AVESKIP = 2^SKIP

PIXELSTEP:

MAXSTEP = 256 (or 65536 if more precise pixel positioning is desired)

PIXELSTEP = Floor((MAXSTEP * AVESKIP) / IR)

Calculation example, ITR = 5,5:

SKIP = Floor (LOG2(ITR)) = Floor (2,46) = 2

AVESKIP = 2^2 = 4

PIXELSTEP = Floor (256 * 4 /5,5) = 186

In the case of Figure 4c, it is sought to set the first and second-stage scaling ratios to be equal, in which case 1/X is approximately Y/Z. The memory and processing requirements will then be reasonable and the image quality nearly optimal.

The integer calculation auxiliary variables AVESKIP and PIXELSTEP are defined in the following:

AVESKIP:

Inverted total scaling ratio IR = MAX(Hin / Hout, Vin / Vout), in which horizontal (H) and vertical (V) sizes are used.

AVESKIP = Floor(Sqrt(IR))

PIXELSTEP:

MAXSTEP = 256 (or 65536, if more precise pixel positioning is desired)

PIXELSTEP = Floor(MAXSTEP * AVESKIP) / IR)

Calculation example 3, ITR = 5,5:

AVESKIP = Floor (Sqrt (5,5)) = 2

MAXSTEP = 256

PIXELSTEP = Floor (256 * 2 /5,5) = 93

In cases in which the total scaling ratio is on 1/X, AVESKIP=X, the second stage is bypassed.

In most applications, scaling is made directly to the sensor circuit, using a suitable processor unit. The silicon area of such a circuit (ASIC) can be reduced to a usable level using a scaling method disclosed above, compared to an optimal scaling method. With the aid of the invention, it is possible to minimize the amount of line memory on the chip. This is because the area of the silicon determines the costs of the chip and is a central cost factor in portable

camera solutions. The invention permits smaller cameras than previously and an increased dynamic range. The coded image size is smaller for the same quality parameters and is suitable for both separate images and video images.

The examples above are mainly for hardware-implemented camera sensors, but the invention can also be applied outside of the sensors, for example, by means of software in a PC. The camera sensor is suitable for use, not only in actual cameras, but also in mobile telephones (generally in mobile stations).